

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Rok Ivančič

**Senzorski modul za spremljanje
hladne verige**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM
PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Mira Trebar

Ljubljana, 2017

COPYRIGHT. Rezultati diplomske naloge so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavo in koriščenje rezultatov diplomske naloge je potrebno pisno privoljenje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Številni senzorji predstavljajo vse bolj pomembno vlogo v sistemih, ki jih srečujemo v povezavi z internetom stvari. Omogočajo zbiranje podatkov o okolju v katerem se hranijo živila pri katerih je pomembno zagotavljanje hladne verige. Kandidat naj zasnuje in implementira senzorski modul, ki povezuje senzorje temperature, vlage in plinov v samostojno napravo tako, da uporabi odprto kodno ogrodje Arduino za pošiljanje podatkov na mikroračunalnik Raspberry Pi. Delovanje zasnovanega prototipa naj predstavi z rezultati meritev v grafu za nekaj različnih primerov.

Na tem mestu bi se rad zahvalil mentorici doc. dr. Mira Trebar, ki mi je dodatnimi komentarji pomagala pomagala pri izdelavi diplomskega dela. Velika mera zahvale gre tudi mami in bratu ker sta me vzpodbujala skozi celoten študij ter puncu, ki mi je vedno stala ob strani.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Opis področja	3
2.1	Internet stvari	3
2.2	Hladna veriga	4
2.3	IoT v hladni verigi	6
3	Senzorski sistem	9
3.1	Strojna oprema	10
3.2	Programska oprema	18
3.3	Arhitektura	20
4	Izvedba in primer uporabe	23
4.1	Senzorski sistem - prototip	23
4.2	Primer uporabe	34
5	Sklepne ugotovitve	41
	Literatura	42

Seznam uporabljenih kratic

kratica	angleško	slovensko
CO₂	Carbon dioxide	Ogljikov dioksid
GPIO	General-purpose input/output	Vhod / izhod za splošne namene
HDMI	High Definition Multimedia Interface	Multimedijski vmesnik visoke frekvence
IDE	Integrated Development Environment	Vgrajeno razvojno okolje
IDLE	Integrated Development and Learning Environment	Integrirano razvojno in učno okolje
IoT	Internet of Things	Internet stvari
ISM	Industrial, scientific and medical	Industrija, znanost in medicina
LED	Light-emitting diode	Svetleča dioda
SD	Secure Digital	Digitalno varna
SPI	Serial Peripheral Interface Bus	Serijski periferni vmesnik
USB	Universal Serial Bus	Univerzalni podatkovno vodilo

Povzetek

Naslov: Senzorski modul za spremljanje hladne verige

Internet stvari se pojavlja na številnih področjih, kjer je pomembno, da naprave komunicirajo med seboj in so povezane v internetnem omrežju. Glavni namen njegove uporabe je poenostavljeno opravljanje raznovrstnih nalog tako v industrijskem, kot tudi v domačem okolju. V diplomski nalogi je predstavljena rešitev za spremljanje in analizo hladne verige s senzorskim sistemom za zajem in pregled podatkov. Uporabljena sta dva mikrokrmilnika Arduino in Raspberry Pi, ki sta trenutno med najbolj priljubljenimi. Pravzaprav sta to majhna računalnika, ki sestavljata prototip realiziranega senzorskega sistema. Arduino ima priključene senzorje za temperaturo, vlago ter gorljive pline in omogoča, da se ves čas delovanja izvajajo meritve in zbirajo podatki. Preko radio-frekvenčnega oddajnika ter sprejemnika jih nato pošilja na Raspberry Pi. Naloga slednjega je sprejemati podatke in jih s pomočjo enostavne aplikacije prikazati ali pa poslati na izbran elektronski naslov. Senzorski sistem omogoča spremljanje stanja ozračja (temperatura, vlaga, plini) v prostoru ali na prostem, kjer je bilo izvedeno tudi testiranje. Na podanih grafih je predstavljeno tudi spremljanje temperature in vlage v spremenljivih vremenskih razmerah ter zaznavanje plina (CO₂).

Ključne besede: Internet stvari, Arduino, Raspberry Pi, temperatura, vlaga, gorljivi plini

Abstract

Title: Sensor module for cold chain monitoring

Internet of things appears in many fields where it's important that devices communicate to each other and are connected to the Internet. The main purpose of its use is to simplify various tasks in the industrial and home use. The diploma thesis presents a solution for monitoring and analyzing a cold chain with sensor system to capture and review collected data. Two microcontrollers, Arduino and Raspberry Pi are used, which are currently the most popular. In fact, these are two small computers that construct the prototype of the realized sensor system. Arduino has connected sensors for temperature, humidity and flammable gases and provides carrying out measurements during the system operation and collects data. By using radio-frequency transmitter and receiver it sends them to Raspberry Pi which receives data and using a simple application displays or sends them to selected e-mail address. Sensor system enables monitoring of the state of the atmosphere (temperature, humidity, gases) in the room or outdoors where testing has also taken place. On graphs it is also presented monitoring of temperature and humidity in various weather conditions and detection of gas (CO₂).

Keywords: Internet of things, Arduino, Raspberry Pi, temperature, moisture, flammable gasses

Poglavje 1

Uvod

Hladna veriga je v osnovi pereč problem, saj se lahko stanje določenega živila med transportom in skladiščenjem zelo spremeni. Živilu se zaradi neprilagojenega prostora, v katerem se nahaja med transportom, zmanjša hranilna vrednost ali pa izgubi svoj prvoten namen. Te težave se pojavljajo pri transportu cepiv, sadja, zelenjave, zamrznjene hrane, raznih mlečnih izdelkov, itd. Transport se smatra kot čas in prostor, ki ga živilo prepotuje od proizvodnje do končnega uporabnika, zato je smiselno, da se stvari lotimo na drugačen način.

Zahvaljujoč tehnologiji nam Internet stvari - IoT (Internet of Things) omogoča spremljanje, analiziranje in izvajanje določenih ukrepov pri živilih, katere želimo nadzirati. Ta nam pomaga, da pridobimo informacije o transportu in ozračju, kjer se živilo nahaja. Tako lahko vidimo kje je temperatura ozračja neprimerna in na podlagi tega podatka ukrepamo. V ta namen se s pomočjo senzorskega sistema spremlja temperaturo, vlago in pline, ki vplivajo na kakovost živil v transportu. Cilj je sestaviti sistem, ki bo v prostoru s pomočjo senzorjev zbiral podatke, jih urejal, shranil in predstavil uporabniku za kasnejšo analizo. V ta namen se s pridobljenimi podatki analizira učinkovitost in ugotovi šibke točke transporta. Na ta način se lahko hitreje in bolj uspešno odzivamo in zagotovimo živilu primeren transport, posledično pa boljšo kakovost in daljšo življenjsko dobo, katerih korist imajo na koncu

potrošniki.

Cilj diplomskega dela je zasnovati in izdelati enostaven, cenovno ugoden IoT sistem (Arduino, Raspberry Pi), ki bo iz okolja neprekinjeno, preko senzorjev, pridobival trenutne podatke o temperaturi, vlagi in prisotnosti plinov ter nato te podatke posređoval uporabniku preko elektronske pošte v obliki razpredelnice. V senzorskem sistemu so vključeni senzorji za merjenje temperature, prisotnosti vlage in plinov. Za prenos podatkov iz senzorskega sistema na aplikacijski je uporabljen radio frekvenčni oddajnik in sprejemnik.

Diplomsko delo vsebuje kratko predstavitev IoT, opis hladne verige, ter načrta za izdelavo senzorskega vozlišča. Tretje poglavje se navezuje na uporabljeno strojno in programsko opremo, ter opis uporabljenih komponent. Izvedba in delovanje sistema sta opisana v četrtem poglavju, kateremu sledi še primer uporabe.

Poglavje 2

Opis področja

Internet stvari (IoT) je opisan z vidika splošne uporabnosti, ker dobiva vse večji pomen tudi v vsakdanjem življenju človeka. V nadaljevanju sledi kratka predstavitev hladne verige in problemov, ki so vezani na to področje, ter kako lahko IoT z različnimi senzorskimi sistemi ponudi enostavne in uporabne rešitve na tem področju.

2.1 Internet stvari

Internet stvari, ni več povsem novo tehnološko področje. Podjetja in strokovnjaki že desetletja razpravljajo o ideji povezljivosti naprav tako, da si med seboj izmenjujejo podatke oziroma komunicirajo. Tako je prva naprava, ki je bila povezana v internet, preprosti taoster, predstavljen na konferenci leta 1989 [17].

Pomen IoT je preprost. Gre za povezovanje naprav preko interneta, tako da se pogovarjajo z nami, aplikacijami in med seboj. Najbolj priljubljen, sicer nenavaden primer je pameten hladilnik. Ideja je, da nam hladilnik sporoči, če je zmanjkalo mleka, masla ali kateregakoli drugega živila tako, da nam pošlje obvestilo preko sms sporočila ali pa nas opozori, da je živilu pretekel rok uporabe

Najpogostejša uporaba IoT v stanovanjskih hišah se, vsaj v Združenem

kraljestvu [17], uporablja za ogrevanje in porabo energije. Delno tudi zato, ker vlada spodbuja energetska podjetja, da uvedejo pametne števec. Ti imajo pametne funkcije, ki omogočajo da se z daljinskim upravljanjem zviša temperaturo, zniža, če je sončni dan ali celo izklopi, ko nas ni doma. Nekateri števeci lahko preko senzorjev za zaznavanje gibanja ugotovijo, da nismo več v prostoru ali pa preprosto zaznajo, da je pameten telefon (in zato tudi mi) zapustil prostore.

Področje uporabe IoT je več kot le pametna hiša povezanih naprav. Uporabnost lahko sega tudi v velika mesta, torej pametna mesta. Idej je veliko, lahko govorimo o povezanih prometnih signalih, ki nadzirajo pretočnost prometa ali pri spremljanju komunalnih storitev oziroma pametnih zabojsnikov za smeti, ki signalizirajo, ko jih je treba izprazniti. Ne smemo pozabiti na vso industrijo in z njo povezanimi senzorji, ki lahko spremljajo vse, od sledenja delov do spremljanja dogajanja na poljih.

Zakaj je to pomembno? Obstaja razlog, da Britanska vlada spodbuja energetska podjetja, ki ponujajo pametni števec. Vsi ti podatki in avtomatizirana uporaba so učinkovitejši, kar pomeni, da porabimo manj energije. Veliko področij IoT kažejo takšne koristi, čeprav so nekateri pametni pripomočki na določenih področjih bolj učinkoviti, drugi manj [7].

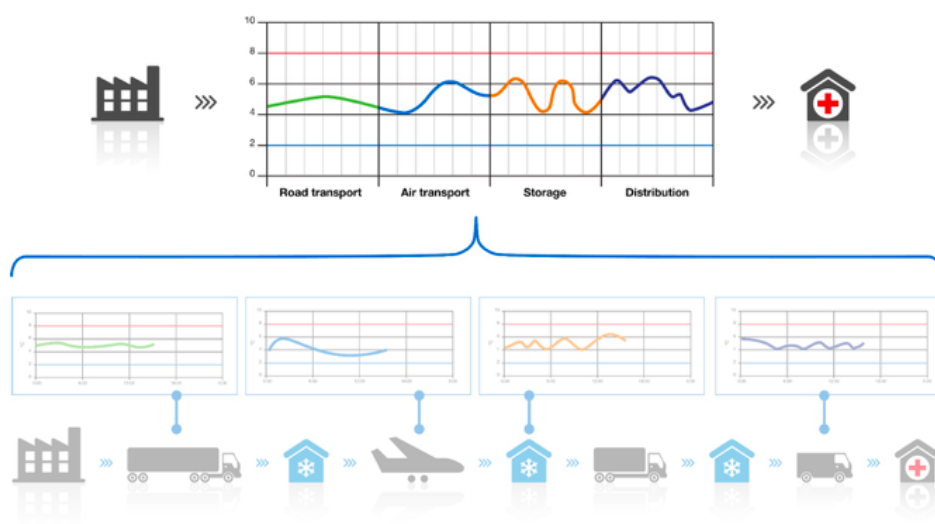
2.2 Hladna veriga

Hladna veriga je pomembna pri nadzoru ravnanja s pokvarljivimi proizvodi od mesta izdelave do končnega uporabnika. "Pomeni vzdrževanje predpisane, ustrezne temperature živila, da ohranimo čim boljše kakovost proizvoda v celotni živilski verigi od proizvodnje, shranjevanja, do prenosa h končnemu potrošniku"[6]. Vsak živilski proizvod ima predpisano optimalno temperaturo za shranjevanje in temperaturno mejo, do katere še ustreza zahtevam za varno uporabo. Veliko živil, ki potujejo v toplejše kraje (npr. cepiva, sadje, zelenjava, zmrznjena hrana,...), prispe pokvarjenih ali neaktivnih, ker niso bili podvrženi zahtevanim pogojem hladne verige. Najpogosteje na-

stane napaka v transportni logistiki, kjer morajo živilo pripeljati na cilj do prodajalcev. Bistvo hladne verige je, da živilo hitro, oziroma v najkrajšem možnem času, naložijo na primerno ohlajeno transportno vozilo in tega brez temperaturnih šokov pripeljejo na cilj. Med izvedbo procesa dostave je lahko transportnih vozil tudi več, kar pomeni še dodatno težavo ali možnost za napake v vzdrževanju primerne temperature.

Z zagotavljanjem in optimizacijo hladne verige, se vse pogostejše uporablja IoT. Dandanes se pri vseh hladilnih komorah, kjer so spravljena živila, dnevno meri temperatura ozračja. Zbirajo se podatki o spreminjanju temperature skozi celoten razvojni in logistični proces. S pridobljenimi podatki lahko analiziramo ter ocenimo ustreznost pogojev v prostorih z živili.

Slika 2.1 prikazuje vse faze nadzora od proizvodnje, do mesta prodaje, kjer potem živilo dobi končni uporabnik. Med samim transportom se izvaja spremljanje temperature v odvisnosti od časa, kjer nam prikazuje kako se temperature spreminjajo ter kje so kritične točke, ki bi škodovala živilu [8].



Slika 2.1: Hladna veriga, transport [8]

2.3 IoT v hladni verigi

Za proizvajalca je pomembno, da proizvod, ki ga ustvari, tudi dostavi v enakem stanju, kot je prišel iz proizvodnje, zato je transport živila ključnega pomena, da zagotavlja primerne pogoje za dostavo. Da se prepričamo o optimalnosti pogojev in da se v prihodnje izognemo morebitnim težavam, prostor nadziramo s senzorji, ki merijo ustrezne pogoje ozračja. Najpomembnejše lastnosti, ki jih je potrebno meriti in so vključene v diplomskem delu so:

- Temperatura
- Vlaga
- Ogljikov dioksid (CO_2)

Lastnosti ozračja se v času transporta lahko bistveno poslabšajo, razlogi za to pa so lahko različni. Če se živila večkrat prestavlja iz enega transportnega vozila na drugega, po možnosti še v krajih s toplejšim ali vlažnim ozračjem, se lahko povečata tako temperatura kot vlaga, ki lahko vplivata na kakovost živila. Ponekod lahko pod določenimi pogoji pride tudi do požara, kar povzroči proizvajalcu ali kupcu visok denarni strošek, zato je smiselno, da se zadeve konstantno meri in analizira, da se prepreči izgube. Seveda obstaja poleg naštetih še vrsta drugih dejavnikov, ki so pomembni za merjenje.

IoT pomaga tako, da meri vrednosti naštetih dejavnikov, ki so v prostoru, posreduje podatke ustreznim organom, po možnosti pa glede na to tudi ukrepa. Prostor, v katerem se nahaja živilo med transportom, je tarča merjenja ozračja s senzorji in drugimi pripomočki, ki nam dajejo povratno informacijo. Glede na podatke, ki jih dobimo, lahko ustrezno reagiramo v korist živila. Podatki se po modulih obdelajo in prenesejo do uporabnika v človeku berljivi obliki. Tako lahko iz pridobljenih podatkov analiziramo potek transporta in ugotovimo, kje so šibke točke. Sistem se lahko razširi do te mere, da v primeru, ko sistem med transportom zazna merjeno količino, ki presega dovoljeno, sproži ustrezno akcijo, ki nas obvesti, da je živilo v ne-

primernem okolju. Ta akcija je lahko alarm, mobilno sporočilo, elektronska pošta ali kaj podobnega.

Poglavje 3

Senzorski sistem

V nadaljevanju je predstavljena arhitektura senzorskega sistema za spremljanje hladne verige. Sledi tudi opis strojne in programske opreme, s katero je izdelan senzorski sistem.

Senzorski sistem je namenjen spremljanju hladne verige in prenosu podatkov do uporabnika. Pri razvoju samostojnega sistema je potrebno poskrbeti za stabilnost in varnost sistema predvsem v brezžični komunikaciji, če so podatki, ki jih prenašamo, namenjeni le določeni osebi. Postavitev sistema, predvsem senzorjev, se od sobe do sobe lahko razlikuje. Pomembno je, da so senzorji v središču dogajanja, aktivnosti in ozračje v sobi pa ne vplivajo ali poškodujejo komponent. Sistem je sestavljen iz dveh delov:

- Senzorski (senzorji, Arduino)
- Strežniški oziroma aplikacijski (Raspberry Pi)

V prvem delu so vključeni senzorji, ki so direktno (žica) povezani na modul Arduino. Na njem teče program za pridobivanje podatkov iz senzorjev, ki jih primerno obdela ter skrbi za njihovo lokalno, neobstoječe shranjevanje, ki se potem v določenih intervalih prenašajo naprej na modul Raspberry Pi. Podatki veljajo za neobstoječe zato, ker se trenutni podatki med izvajanjem nikamor ne zapisujejo, ampak so med čakanjem na prenos shranjeni v Flash pomnilnik, ki bi v primeru nenadne prekinitve izgubil vse podatke.

Količina izgubljenih podatkov je odvisna od velikosti intervala za pošiljanje. Podatki se ob določenem intervalu pošljejo preko brezžične povezave na drugi, strežniški nivo, oziroma modul Raspberry Pi. Ta modul v enakih intervalih kot Arduino čaka na odziv in sprejema podatke. Za ta proces poskrbi majhna namizna aplikacija, ki prikazuje prenos podatkov, ter ima dva gumba za začetek in prekinitev prenosa podatkov. Po prekinitvi sprejemanja se sprejeti podatki pošljejo naprej uporabniku preko elektronske pošte ali pa se samo shranijo v lokalnem pomnilniku.

3.1 Strojna oprema

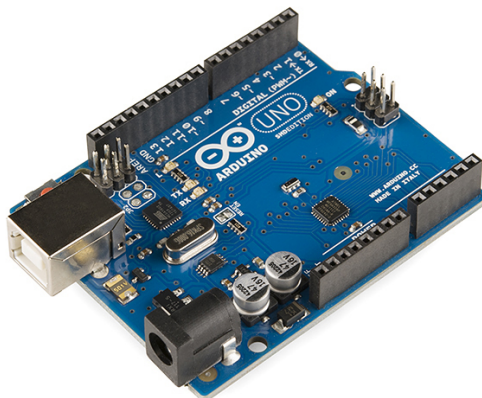
Nabava strojne opreme za Arduino in Raspberry Pi je bila preprosta, saj so vsi senzorji in drugi pripomočki že prilagojeni za ta dva modula. Pomembno je samo, da se za senzorje zagotovi primerno napajanje in se jih pravilno doda v vezje.

3.1.1 Arduino

Arduino je odprtokodna platforma za izdelavo digitalnih projektov. Sestavljen je iz fizičnega programabilnega vezja (pogosto imenovanega mikrokrmilnik) in programske opreme ali IDE (Integrated Development Environment), ki se izvaja na osebнем računalniku. Preko programske opreme lahko napišemo program za Arduino, ki ga preko iste opreme naložimo na fizično ploščo. Platforma Arduino je postala precej priljubljena pri ljudeh, ki se šele začenjajo ukvarjati z elektroniko. Za razliko od večine prejšnjih programirljivih vezij ne potrebuje ločenega dela strojne opreme, da bi na novo naložili programsko kodo, ampak lahko preprosto uporabimo USB kabel. Poleg tega uporablja Arduino poenostavljeno različico programskega jezika C, kar olajša pisanje programov [16].

Arduino je narejen na osnovi 8-bitnega mikrokrmilnika ATmega328P, kjer njegova delovna frekvenca znaša 16 MHz. Ponuja 6 analognih in 14 digitalnih vhodno-izhodnih pinov, ki jih lahko nastavljamo ali pa beremo njihova stanja.

Napetost v vsakem izmed pinov znaša 5 V, tok pa 40 mA. Za pomnjenje programa je namenjeno 32 KB flash pomnilnika. Na sliki 3.1 je prikazan mikrokrmilnik Arduino [15].

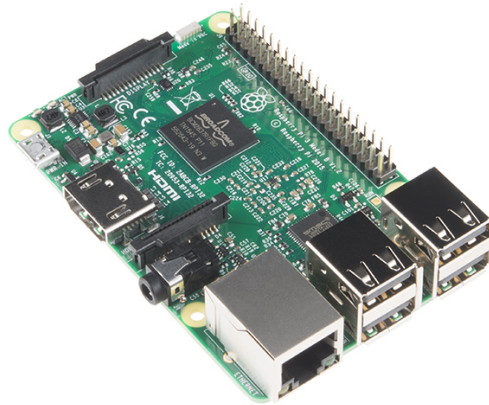


Slika 3.1: Arduino Uno [15]

3.1.2 Raspberry Pi

Raspberry Pi je mikroročunalnik v velikosti kreditne kartice, ki so ga razvili v Združenem kraljestvu za potrebe poučevanja računalništva. Pomembna posebnost je kompaktnost, saj združuje vse elemente stacionarnega računalnika v enem majhen kompletu. Njegova moč je povsem primerljiva s sicer nekoliko starejšimi računalniki, se pa lahko pohvali s štirijedrnim procesorjem quad-core A53 @ 1,2 GHz, 1 GB pomnilnika in ločeno grafično kartico z urinim taktom 400 MHz. Da Raspberry Pi sledi standardom in zahtevam uporabnikov, zato ponuja tudi Wireless 802.11n ter Bluetooth 4.0 tehnologiji, HDMI video priključ, štiri USB vhode, Ethernet vhod, rezo za SD kartico ter 40 vhodno-izhodnih (GPIO) pinov. Največja prednost tega računalnika pa je seveda to, da nanj lahko naložimo poljuben operacijski sistem. Prvotno je podpiral različne distribucije Unix / Linux operacijskega sistema, v ta namen

je bil razvit operacijski sistem Raspbian, poleg pa se je vmešal tudi Microsoft s svojim operacijskim sistemom Windows 10 Lite oziroma Windows 10 IoT Core. Raspberry Pi za shranjevanje podatkov uporablja SD kartico, kamor tudi naložimo operacijski sistem. Slika 3.2 prikazuje mikroračunalnik Raspberry Pi Model 3 B [9].



Slika 3.2: Raspberry Pi Model 3 B [9]

3.1.3 Senzor za temperaturo

Za spremljanje temperature ozračja se uporablja senzor DS18B20, ki je povsem plastificiran in vodoodporen. Z njim lahko merimo temperaturo ozračja, tekočine ali zemlje. Zasnovan je s tremi žicami. Ker deluje po principu enožičnega protokola, uporablja samo eno žico za pošiljanje podatkov o temperaturi. Poleg sta še dve drugi, ena se uporablja za napajanje senzorja in druga za ozemljitev. Območje za napajanje senzorja znaša od 2.7V - 5.5V. Temperaturo lahko meri v območju od -55°C pa do $+125^{\circ}\text{C}$ z natančnostjo $\pm 0.5^{\circ}\text{C}$ med -10°C in 85°C . Senzorju lahko še dodatno nastavljamo natančnost oziroma ločljivost med 9 in 12 biti, vendar je ta pogojena s časom pridobivanja podatkov. Manjša kot je ločljivost, manjša je razlika

med temperaturami (0.5°C , 9-bitna ločljivost), hitrejši je zajem podatkov (≈ 93 milisekund), oziroma večja kot je ločljivost ($0,0625^{\circ}\text{C}$, 12-bitna), večji je čas zajema podatkov (≈ 750 milisekund). Senzor sporoča podatke tako, da preko analognega vhoda pošilja vrednost napetosti glede na temperaturo v ozračju. Če je temperatura visoka, bo vrnjena napetost v Arduino tudi visoka in obratno. Slika 3.3 prikazuje senzor za temperaturo [14].



Slika 3.3: Senzor za temperaturo, DS18B20 [14]

Kot omenjeno, za napajanje lahko uporabljamo napetost 2.7V ali 5.5V . V vsakem primeru bo izhodna analogna napetost med 0V (ozemljitev) in 1.75V . Da pravilno preberemo temperaturo iz analognega pina, uporabimo matematično formulo, ki nam pretvori 10-bitno analogno vrednost v temperaturo. Formula

$$\text{pinVoltage} = \text{ADC} * (5000/1024)$$

kjer *pinVoltage* pomeni napetost pina v milivoltih in *ADC* dejanska napetost na pinu, pretvori število med $0 - 1023$ iz analognega pina v $0 - 5000\text{mV}$,

če za vhodno napetost uporabljamo 5V. Nato za pretvorbo milivolte v temperaturo, uporabimo formulo

$$Temp = [pinVoltage - 500]/10$$

kjer je *pinVoltage* prej izračunana vrednost, *Temp* pa dejanska temperatura (°C) v prostoru [14].

3.1.4 Senzor za vlago

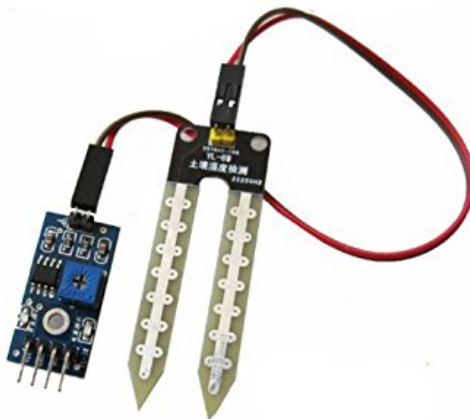
Med pomembnimi dejavniki je tudi prisotnost in koncentracija vlage v ozračju. V ta namen se za merjenje vlage uporablja za to namenjen senzor FC-28. Ta meri prostorninsko vsebovanost vode v tleh ali ozračju in nam daje raven vlage. Senzor vlage je sestavljen iz dveh sond, ki se uporabljata za merjenje prostorninske vsebovanosti vode. Sonde omogočata pretok toka skozi tla ali ozračje, iz česar dobimo vrednost upora, ki nam pove vrednost vlage. Ko je voda prisotna, tla ali ozračje prevajata večjo količino elektrike, kar pomeni, da bo ob višji ravni vlage upora manj, suha tla ali ozračje pa slabo prevajata elektriko, zato bo v takem primeru več električnega upora. Razpon napetosti, ki ga vrača senzor, je od 0.1 do 4.2V. Tako lahko glede na vrnjeno napetost izrazimo vlago v ozračju kot procent [3].

Količino prisotnosti vlage v ozračju merimo v procentih (%). Izhodna napetost senzorja na analognem pinu znaša med 0 - 1023, zato delež dobimo s formulo

$$odstotek = analogRead/1024$$

kjer je *odstotek* količina oz. delež vlage v prostoru, *analogRead* pa izhodna vrednost senzorja.

Senzor (Slika 3.4) za delovanje uporablja štiri žičke. Ena je za napetost (3.3 V - 5 V), ena za ozemljitev in ostali dve za digitalni ali analogni izhod. Poleg je senzor opremljen še z potenciometrom, ki določa natančnost preverjanja prisotnosti vlage.



Slika 3.4: Senzor vlage, FC-28 [3]

3.1.5 Senzor za gorljive pline

MQ-2 Gas senzor je med najbolj priljubljenimi senzorji za spremljanje gorljivih plinov. MQ tipi senzorjev v notranjosti vsebujejo majhen grelec z elektrokemičnim senzorjem, ki zaznava serijo gorljivih plinov, kot so utekočinjen naftni plin, butan, propan, metan, alkohol, vodik in CO₂. Grelec se v času delovanja močno segreje, zato se ga v času delovanja ne smemo dotikati. Senzor deluje tako, da spreminja izhodno vrednost napetosti glede na prisotnost dima in/ali plina v ozračju. V primeru visoke koncentracije plina, senzor pošilja večjo napetost in obratno, manjša kot je koncentracija plina, manjša vrednost napetosti je na izhodu [11]. Senzor vsebuje štiri žičke, eno za napetost (najprimernejša napetost je 5V), drugo za ozemljitev in dve za pošiljanje podatkov, analogni in digitalni izhod. Analogni izhodni signal iz senzorja se lahko prebere z analognim vhodom Arduino modula. Poleg tega ima še dodatne komponente, kot je potenciometer, s katerim lahko določamo natančnost zajema podatkov, ter LED lučke, ki gorijo v primeru delovanja senzorja. Senzor je prikazan na sliki 3.5.

Posebne enote za prikaz količine prisotnosti plinov nimamo, zato to količino

predstavljamo z deležem oziroma procenti (%). Izhodna napetost senzorja na analognem pinu znaša med 0 - 1023, zato delež dobimo s formulo

$$\text{odstotek} = \text{analogRead}/1024$$

kjer je *odstotek* količina oz. delež plinov v prostoru, *analogRead* pa izhodna vrednost senzorja [12].



Slika 3.5: Senzor gorljivih plinov, MQ-2 [12]

3.1.6 Oddajnik-Sprejemnik

Za prenos podatkov med moduloma Arduino in Raspberry Pi se uporablja brezžični modul nRF24L01, Slika 3.6. Gre za modul, ki lahko oddaja ali sprejema podatke, torej deluje obojestransko. Uporabljeni sta dva takšna modula, eden za Arduino, ki pošilja podatke, ter drugi za Raspberry Pi, ki podatke sprejema. Gre za serijo nizkocenovnih radiofrekvenčnih modulov, ki so osnovani na Nordic Semiconductor nRF24L01+ čipu. Nordic nRF24L01+ je visoko integriran, nizko potraten, 2 Mbps radio frekvenčni sprejemnik za 2.4 GHz ISM (Industrial, Scientific and Medical) pas. Podpira SPI vmesnik z visokimi hitrostmi za aplikacijsko upravljanje.

SPI (angl. Serial Peripheral Interface Bus) je standard, ki opisuje sinhrono serijsko povezavo med različnimi elektronskimi napravami (integrirana vezja). Deluje v dvosmernem (angl. Duplex) načinu. Za razvoj tega je odgovorno podjetje Motorola, ki ga je uspelo razviti v 80-ih leti 20. stoletja. Za komunikacijski protokol uporablja princip gospodar-podrejen (angl. master/slave), kjer nadrejena naprava (gospodar) vzpostavi stik in narekuje komunikacijo s podrejeno napravo.

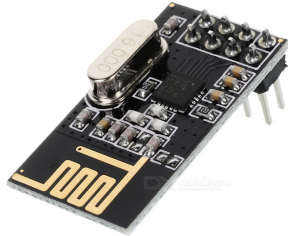
Pomembna lastnost vmesnika SPI je dvosmerna komunikacija s podrejenimi napravami. V ta namen uporablja štiri linije po katerih potuje signal:

- MISO - gospodar posluša, podrejeni sporoča (angl. master input - slave output)
- MOSI - gospodar sporoča, podrejeni posluša (angl. master output - slave input)
- SCK - takt ure, ki ga definira gospodar
- SS - signal, s katerim gospodar določi katera podrejena naprava bo z njim komunicirala.

Z vodenjem SPI lahko gospodar komunicira z samo eno podrejeno napravo. Ta standard lahko uporablja en gospodar in več podrejeni naprav.

NRF24L01 uporablja frekvenčni pas 2.4 GHz in lahko deluje s hitrostjo prenosa podatkov od 250 kbps do 2 Mbps. Če se modul uporablja v odprtem prostoru in z nižjo hitrostjo prenosa podatkov, lahko modula komunicirata tudi na 100 metrski razdalji. Vsak modul lahko uporablja 125 različnih kanalov za komunikacijo, tako imamo lahko 125 neodvisnih omrežji na skupni lokaciji. Vsako omrežje torej komunicira na svoji frekvenci od 2.401 GHz, 2.4002 GHz,... do 2.525 GHz. Vsak komunikacijski modul ima lahko 6 nalogov, to pomeni, da lahko komunicira s šestimi drugimi moduli [2].

Poraba modula znaša približno 12mA med prenosom sporočil, kar je celo manj od LED diode. Zahtevana napetost za delujočo napravo pa je med 1.9 do 3.6V.



Slika 3.6: Komunikacijski modul, nRF24L01 [2]

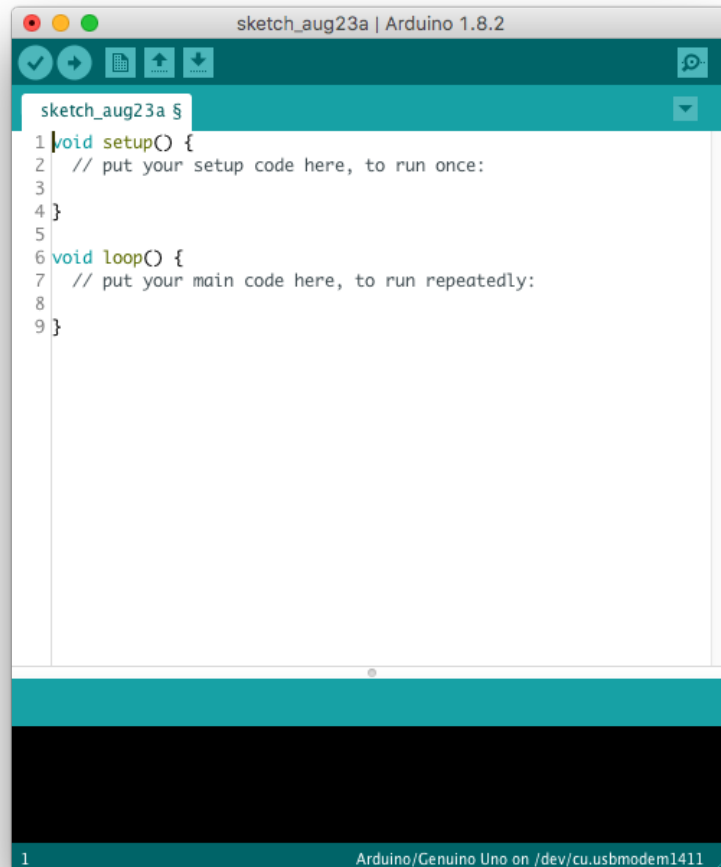
3.2 Programska oprema

Za razvoj Arduino aplikacije se uporablja temu namenjeno integrirano razvojno okolje Arduino, ki je napisano v programskem jeziku Java [1]. Gre za enostaven urejevalnik kode, ki podpira razvoj v programskih jezikih C in C++. V njem so ob namestitvi že naložene določene knjižnice, ki poenostavljajo pisanje programa. Za razvoj projekta je vključena tudi uporaba knjižnic OneWire in DallasTemperature.

Vsak projekt, napisan v tem razvojnem okolju, se imenuje skica (slika 3.7) in se lokalno shrani na računalnik s končnico *.ino*. Skica, napisana v tem razvojnem okolju, sestoji iz dveh glavnih funkcij:

- `setup()`: Ta funkcija se kliče samo enkrat, in sicer samo na začetku, ko se program zažene. Njena glavna naloga je, da inicializira spremenljivke, vhodno-izhodne enote in druge knjižnice, uporabljene v skici.
- `loop()`: Ko se `setup()` funkcija izvede, se zažene funkcija `loop()`, ki se neprestano izvaja in se ponavlja, dokler ni razvijalna ploščica ugasnjena ali ponovno zagnana [1].

Za razvoj aplikacije na ploščici Raspberry Pi se uporablja programski jezik Python 3, katerega knjižnice so posebej napisane in prilagojene za ta računalnik. Prav tako je pisanje kode v programskem jeziku Python nekoliko lažje kot v C-ju. Razvojno okolje IDLE je bilo namenjeno pisanju aplikacije



Slika 3.7: Integrirano razvojno okolje Arduino

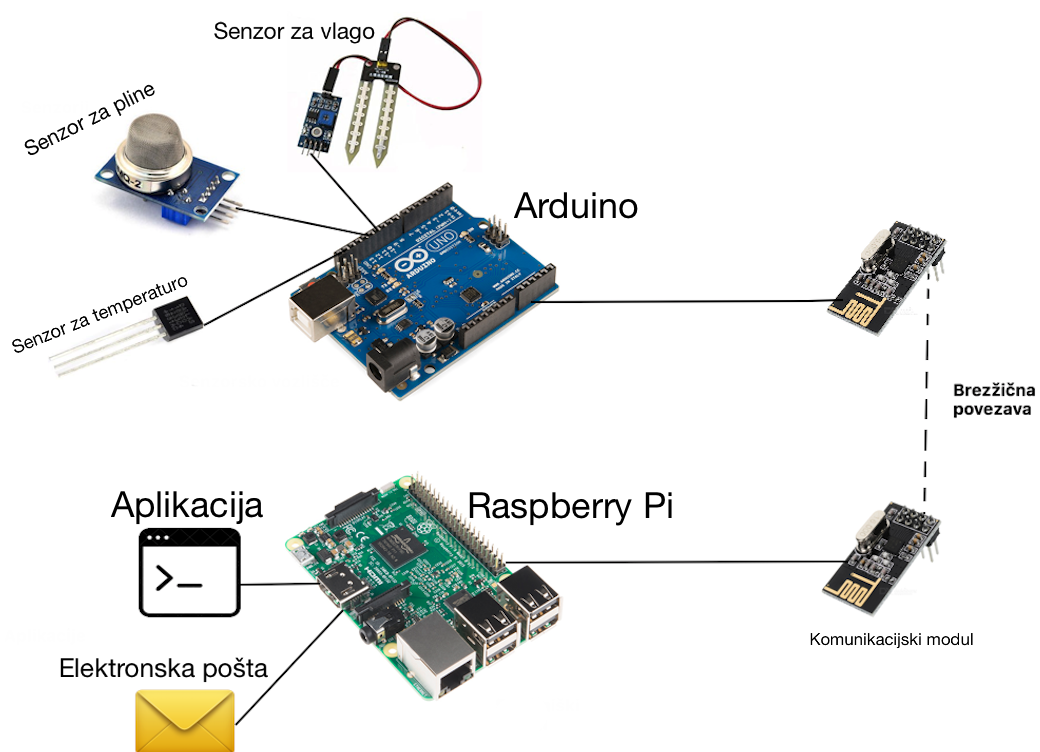
za prikaz podatkov. Razvojno okolje je že bilo nameščeno na operacijskem sistemu in je povsem preprosto, brez posebnih funkcionalnosti razen obarvanja določenega tipa kode. Za razvoj na tem delu projekta se uporablja tudi določene knjižnice, ki olajšajo razvoj programa. To so RPi.GPIO, lib_nrf24, tkinter in MIMEText.

3.3 Arhitektura

Senzorski sistem je ločen na dva dela, katera sta en brez drugega ne uporabna. Prvi (senzorski) del sestavlja Arduino mikrokontroler in trije senzorji, ki pridobivajo podatke. Uporabljeni senzorji so:

- Senzor za temperaturo (DS18B20)
- Senzor za vlago (FC-28)
- Senzor za pline (MQ-2)

Z žicami so vsi omenjeni senzorji povezani na Arduino, na katerem je napisan program v programskem jeziku C, ki upravlja njihovo delovanje tako, da izhodno vrednost senzorjev pretvori v nam razumljivo obliko. Senzorski sistem je samostojen del, ki za delovanje potrebuje le primerno napajanje preko polnilca ali baterij. Njegovo vlogo pa brez drugega (aplikacijskega) dela ne pride do izraza. Drugi del vključuje Raspberry Pi, ki je namenjen sprejemanju, prikazovanju in pošiljanju podatkov. Za komunikacijo med moduloma je uporabljen radio frekvenčni sprejemnik in oddajnik, ki sta oba preko žic povezana na vsak svoj mikrokontroler. Oddajnik na senzorskem delu vsako sekundo zajame podatke, ki mu jih program pridobi in jih preko določenega komunikacijskega kanala pošilja naprej. Na drugi, aplikacijski, strani je sprejemnik, kateri prek istega kanala posluša in zajema podatke, ki mu jih oddajnik pošlje. Zaradi večje količine podatkov in hitrejšega prenosa, se podatki pred pošiljanjem pretvorijo v Bajte, ki zavzamejo manj prostora in se posledično tudi hitreje pošiljajo v radio-frekvenčnem področju. Aplikacija na mikrokontrolerju Raspberry Pi podatke pretvori nazaj v nam razumljivo obliko ter jih izpiše v njenem oknu. Napisana je v programskem jeziku Python in je namenjena spremljanju prejetih podatkov in pošiljanju teh preko elektronske pošte. Arhitektura celotnega senzorskega sistema je predstavljena na sliki 3.8.



Slika 3.8: Arhitektura senzorskega sistema

Poglavje 4

Izvedba in primer uporabe

Razvoj je potekal v dveh fazah. Implementacija senzorskega sistema in testiranje posamezne komponente ter celotnega sistema skupaj z vsemi komponentami. S primerom uporabe pa se lahko vidi rezultate in analizo razvitega sistema.

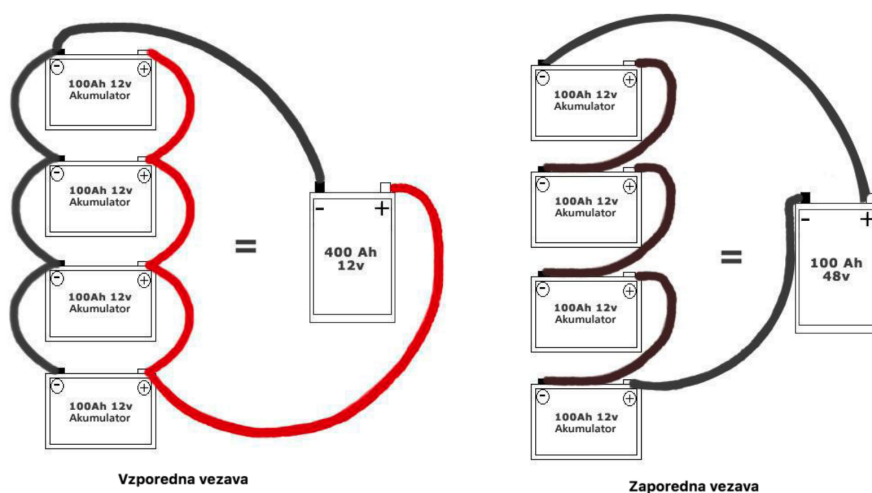
4.1 Senzorski sistem - prototip

Glavni namen senzorskega sistema je, da uporabniku podamo informacije o ozračju v prostoru z živili. Sistem mora biti zasnovan tako, da se uporabnik ne ukvarja z delovanjem, ampak samo pridobi podatke, ki jih sistem vrača. Ta mora delovati brezhibno in biti enostaven za uporabo.

4.1.1 Napajanje

Namen razvitega senzorskega dela sistema je, da ga z vsemi pripadajočimi senzorji postavimo v prostor, kjer želimo, da se opravljajo meritve, zato je potrebno poskrbeti tudi za mobilnost. Tu je najbolj pomembno napajanje modula na lokaciji, kjer je dostop do električne energije omejen. Podoben problem imamo doma, saj ne moremo za vsak senzor napeljati električnega omrežja do naprave ali senzorja. Ta problem rešimo z baterijami, ki so neodvisne od domače napeljave električnega omrežja. Pri odločanju, ko se

zatečemo k akumulatorju, pogosto zanemarjamo razmerja med njihovo zmogljivostjo (navadno izražena v mAh) in močjo, ki jo zahteva plošča Arduino ter z njim povezanimi vezji. To posledično vodi do napačnih ali neustreznih rezultatov, saj je avtonomija delovanja zelo nizka ali pa se sistem niti ne vklopi. Zato se je treba dobro pozanimati, kakšne baterije je potrebno uporabiti, da bo delovanje modula nemoteno in da bo avtonomija primerna za normalno rabo. Ker ena sama baterija ne more vedno zadovoljiti vseh potreb po oskrbi z električno energijo, predvsem zaradi nizke napetosti ali nizke kapacitete, postane pomembno razumevanje zaporednih in vzporednih vezav. To sta načina, na katera sta lahko med seboj povezani dve bateriji za povečanje navedenih vrednosti. Osnovna predpostavka mora biti zelo jasna, da morajo biti vse baterije, ki bodo ustvarile "paket", absolutno enake in morda celo iz iste serije. V primeru alkalnih baterij morajo biti te nove, pri polnilnih baterijah pa morajo biti vse popolnoma napolnjene ali prazne. Na sliki 4.1 je prikazana vzporedna in zaporedna vezava baterij [5].



Slika 4.1: Vzporedna in zaporedna vezava

Vzporedno vezane baterije ohranjajo enako nominalno napetost ene same in povzemajo zmogljivost. Štiri baterije (100Ah, 12V) sestavljajo 12 V baterijo z 400Ah. Ta sistem je uporaben, ko je napetost enojne baterije dovolj za napajanje tokokroga, vendar je potrebna večja avtonomija. Pri zaporedno vezanih baterijah pa se sešteje napetost baterij, a je kapaciteta vezanih baterij enaka eni sami.

4.1.2 Senzorski del

Ker gre za prototip izdelka, je temu namenjena in izbrana tudi strojna oprema. Arduino modul je bila logična izbira za ta projekt, saj ponuja enostavno implementacijo z veliko pomoči preko spleta. V načrtovanju, je bil poleg tega izbran tudi Raspberry Pi, ki velja še za malenkost zmogljivejši oziroma svojemu namenu bolj prilagojen produkt. Poleg sodijo še ostali senzori in drugi pripomočki za izdelavo senzorskega sistema, kot so spajkalnik in multimeter. Pri implementaciji in razvoju smo se najprej osredotočili na delovanje posameznih komponent, ki smo jih potem sestavili v celovit senzorski sistem.

Prvi je bil na vrsti senzor, ki meri temperaturo. Kot smo opisali v prejšnjem poglavju, gre za senzor DS18B20, ki je povsem pastificiran in voodoporen. Arduino je preko žičk priklopljen na razvijalno ploščico tako, da je na pozitivno linijo razvojne ploščice iz mikrokrmilnika pripeljanih 5V in ozemljitev. Tako je razvojna ploščica že pod napetostjo, če bi vklopil Arduino. Sledi temperaturni senzor, ki za vhod dobi napetost 5V. Za pravilno logično raven signala stoji vmes tudi pull-up upor $4.7K\Omega$, ki zagotavlja, da senzor dobiva prave vhodne napetosti ter da nazaj pošilja prave podatke o temperaturi.

Naslednji korak je pisanje programa, ki prebere vrednost iz temperaturnega senzorja in jo prikaže uporabniku. Za pisanje programa na mikrokrmilniku je, kot rečeno, uporabljeno integrirano razvojno okolje Arduino ter jezik C, ki je prilagojen prav za to rabo. To pomeni, da ima vgrajene funkcije in metode, ki poenostavljajo izpise besedil, naslavljanje objektov na

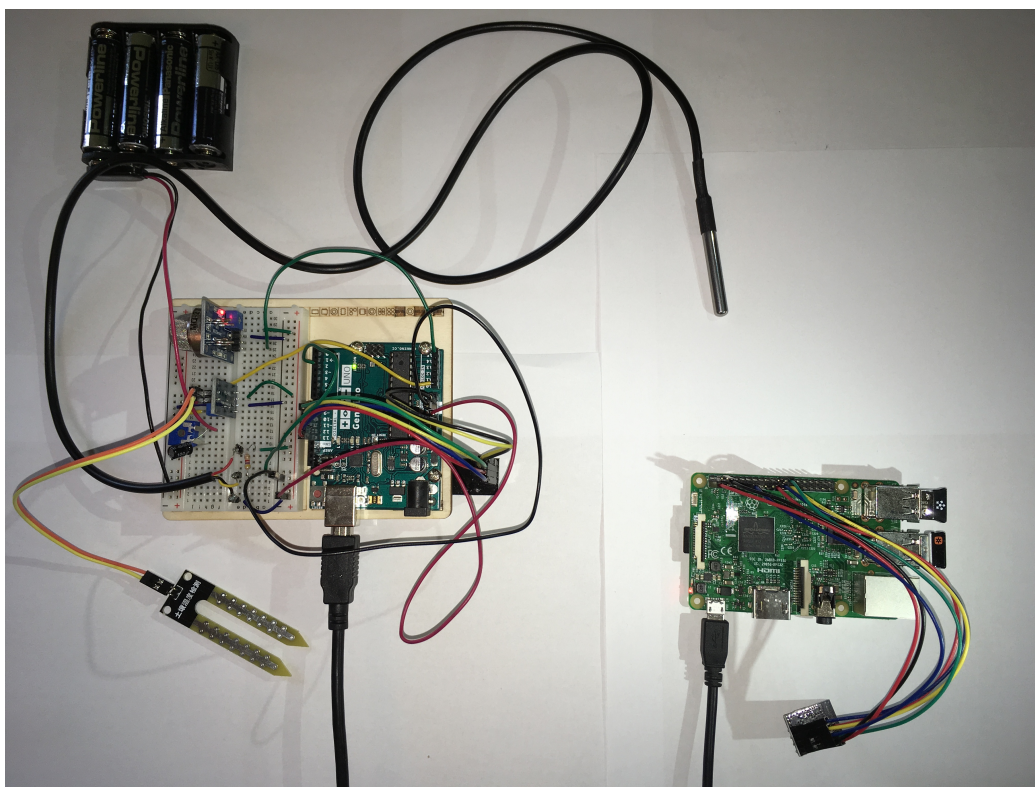
senzorje in podobno. Prav tako ima to razvojno okolje že vključene nekatere knjižnice za lažje komuniciranje s senzorjem. V tem projektu se uporabljata dve knjižnici in sicer OneWire ter DallasTemperature. Knjižnica OneWire se uporablja za komunikacijo med Arduino modulom in senzorjem preko 1-Wire protokola. Za komunikacijo s senzorjem moramo podati le številko pina s katerim je povezan senzor. Knjižnica kreira komunikacijski objekt, katerega reference potem lahko podamo knjižnici DallasTemperature. Ta nam odkrije vse povezane senzorje v tem sistemu, ki nato preko ukaza *getTempCByIndex(0)* vrne izmerjeno temperaturo. V metodi *loop()* je prisotna tudi funkcija *delay(1000)*, ki kot parameter sprejme celo število ter vsak naslednji cikel preloži za naslednjih 1000 milisekund. Za izpis na računalniku se uporablja poenostavljena metoda *Serial.println(temperature)*. Končni rezultat tega dela je bil izpis temperature na vsako sekundo.

Naslednji korak je bil vgradnja senzorja za gorljive pline MQ-2, ki je nekoliko lažji za implementacijo, saj zahteva le točno napetost 5V ter ozeumljitev. Preko analognega pina se zajema napetost, ki jo senzor vrača, ta pa sega med 0.01 in 1.1V. Merskih enot za gorljive pline nimamo točno specificiranih, zato se vrednost vrnjene napetosti pretvoril v procente. To je narejeno tako, da je trenutna izhodna napetost senzorja deljiva z največjo možno. Tako se iz tega izrazi rezultat v procentih (%) in količino prisotnosti gorljivih plinov v ozračju. Težava je pri določanju meje, saj ne vemo, kolikšna je napetost pri kateri je stanje gorljivih plinov že kritično. V ta namen smo z raziskovanjem pri proizvajalčevi shemi ter s testiranjem ugotovili, da je ta meja določena približno pri 400-500mV. V opisu za senzor je omenjeno, da lahko MQ-2 zaznava več plinov, ki jih potem lahko po potrebi filtriramo, da dobimo točno količino vsakega plina. Ker v našem primeru tega nismo vključili kot eno od zahtev, zaznavamo le skupni seštevke plinov. Ko seštevke preseže 500mV oziroma 45% količine gorljivih plinov, pomeni, da je v ozračju nevarnost za vžig. Za implementacijo v programu se uporablja metoda *analogRead(gasPin)*, ki sprejme celo število, ki predstavlja pin

na mikrokrmilniku. Vrednost se nato shrani v spremenljivko, ki gre naprej v obdelavo.

Implementacija naslednjega senzorja je zahtevala nekatere nadgradnje sistema. Oba uporabljena senzorja sta za delovno moč izkoriščala mikrokrmilnik, ki ponuja 3.3 ali 5V izhodno napetost. Ta je bila že v uporabi, zato je bilo potrebno v sistem vključiti zunanji vir energije, ki napaja naš naslednji senzor FC-28 za merjenje vlage. Implementacija senzorja je, podobno kot za MQ-2, enostavna. Za vhodno napetost smo pripeljal zahtevanih 5V ter prek analognega pina brali izhodno napetost senzorja. Kot rečeno pa se za napajanje senzorja uporablja samostojen vir napetosti. Štiri 1.5V baterije vezano zaporedno prinesejo 6V izhodne napetosti. Ker senzor za vlago zahteva $5 \pm 0.1V$, smo morali to napetost s pomočjo potenciometra znižati na 5V za optimalno delovanje senzorja. Enako kot pri senzorju za gorljive pline je izhodna napetost senzorja določena vrednost, ki jo je potrebno pretvoriti v procente za lažje razumevanje tega podatka. Vrednost, ki jo senzor vrača, sega od 0.1 do 4.2V. Tako smo trenutno vrednost delili z najvišjo vrednostjo analognega pina (1024) ter tako določili odstotek vlage v ozračju.

Strojna oprema oziroma senzorski del je s tem korakom končan. To pomeni, da so komponente tako posamično kot tudi kot celota testirane. Podatki, ki jih prejemamo, morajo biti smiselni, da lahko dobimo zaupanje v sistem. Na sliki 4.2 je predstavljen končni senzorski sistem, pripravljen za uporabo.



Slika 4.2: Prototip senzorskega sistema

4.1.3 Brezžična komunikacija

Ker gre za prenosljiv sistem, bi bilo kakršnokoli povezovanje s kabli na Raspberry Pi nesprejemljivo, saj to predstavlja logistični problem, kako postaviti sistem v prostor in z njim upravljati. Podatke, ki jih mikrokontroler preko senzorjev zbere, pošlje naprej v obdelavo na Raspberry Pi, katerega namen je potem podatke prikazati in poslati naprej uporabniku preko elektronske pošte.

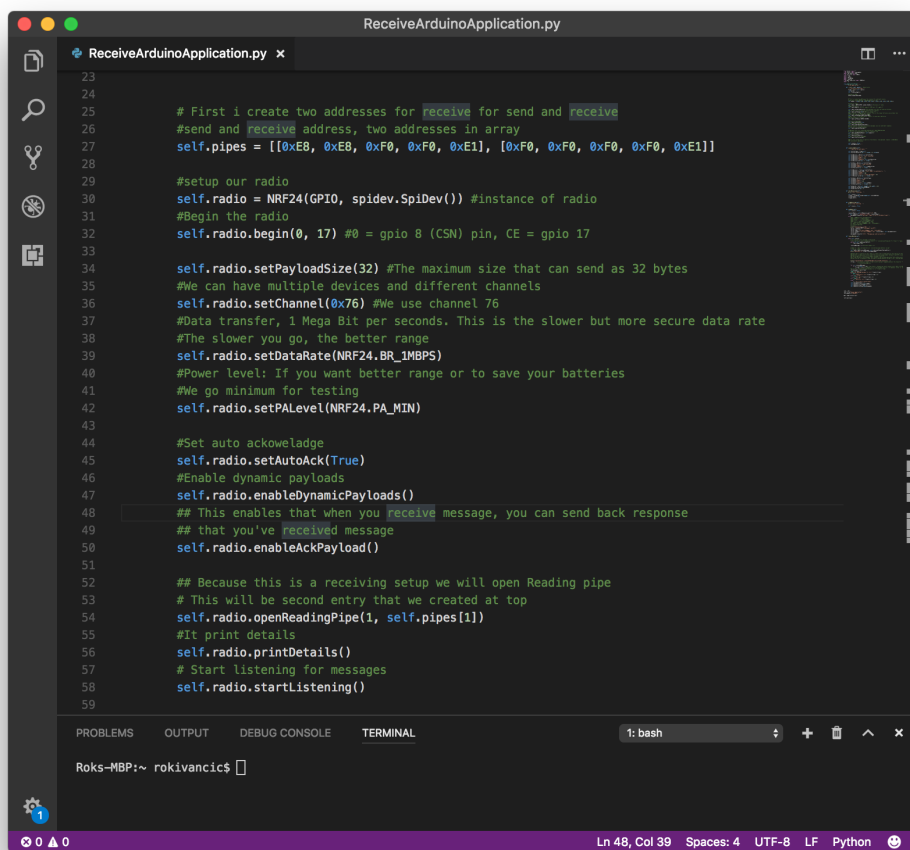
Za komunikacijo med obema mikrokontrolerjema (Raspberry Pi in Arduino) se uporablja modul nRF24L01[18], katerega lastnost je, da lahko opravlja vlogo oddajnika ali sprejemnika. Senzorski sistem vključuje dva komunikacijska modula, za vsak mikrokontroler enega. Za napajanje rabi

modul nRF24L01 3.3V, preko ostalih pinov pa lahko določimo, ali je komunikacijski modul oddajnik ali sprejemnik. Na Raspberry Pi se za pisanje kode za sprejemanje podatkov uporablja programski jezik Python, ki je enostaven za implementacijo tovrstnih modulov ali senzorjev.

Na strani sprejemnika je potrebno nastaviti določene parametre, da lahko vzpostavi komunikacijo ter začne sprejemati sporočila. Glavni parametri, ki jih je potrebno nastaviti, so naslovi za pošiljanje in sprejemanje, določiti je potrebno enolični kanal, preko katerega modula komunicirata, nastaviti modul kot poslušalca ter dati modulu zeleno luč za sprejemanje. Poleg tega lahko nastavljamo tudi lastnosti, kot so *radio.setPayloadSize(32)*, ki pove, da je maksimalna količina poslanih podatkov na enkrat 32 bajtov, ter lastnost *radio.setDataRate(NRDF24.BR1MBPS)*, ki določa hitrost prenosa podatkov na sekundo. V našem primeru se uporablja 1MB na sekundo, kar pomeni počasnejše, ampak bolj zanesljivo prenašanje kot pri višjih hitrostih. Nekaj teh lastnosti je v zapisu kode predstavljenih tudi na sliki 4.3.

Glavni parametri za nastavitve komunikacijskega modula morajo biti identično nastavljeni tudi modulu Arduino. Moč in hitrost pošiljanja lahko poljubno spreminjamo, toda sprejemnik bo sprejemal z najmanjšo hitrostjo, ki je nastavljena na enem ali drugem modulu.

Po uspešno vzpostavljeni komunikaciji pride na vrsto struktura podatkov, ki jih bomo pošiljali. Ker gre za tri tipe podatkov (temperatura, količina gorljivih plinov, količina vlage), se vsakemu podatku na koncu pripne črka, ki označuje tip podatka. Oddajnik podatke pošilja zaporedoma vsako sekundo, kar pomeni, da najprej pošlje na primer temperaturo, potem količino gorljivih plinov, nato še vlago. Na drugi strani sprejemnik sporočilo najprej kodira, nato pa glede na tip izpiše izmerjeno količino. Kako se sporočilo dekodira in izpiše glede na tip podatka je prikazano na sliki 4.4.



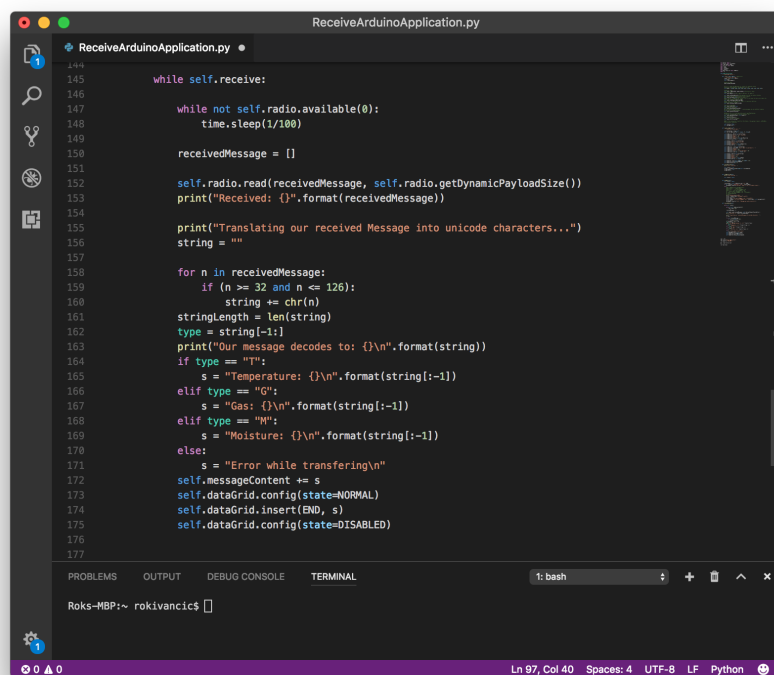
```
23
24
25 # First i create two addresses for receive for send and receive
26 #send and receive address, two addresses in array
27 self.pipes = [[0xE8, 0xE8, 0xF0, 0xE1], [0xF0, 0xF0, 0xF0, 0xE1]]
28
29 #setup our radio
30 self.radio = NRF24(GPIO, spidev.SpiDev()) #instance of radio
31 #Begin the radio
32 self.radio.begin(0, 17) #0 = gpio 8 (CSN) pin, CE = gpio 17
33
34 self.radio.setPayloadSize(32) #The maximum size that can send as 32 bytes
35 #We can have multiple devices and different channels
36 self.radio.setChannel(0x76) #We use channel 76
37 #Data transfer, 1 Mega Bit per seconds. This is the slower but more secure data rate
38 #The slower you go, the better range
39 self.radio.setDataRate(NRF24.BR_1MBPS)
40 #Power level: If you want better range or to save your batteries
41 #We go minimum for testing
42 self.radio.setPALevel(NRF24.PA_MIN)
43
44 #Set auto acknowledge
45 self.radio.setAutoAck(True)
46 #Enable dynamic payloads
47 self.radio.enableDynamicPayloads()
48 ## This enables that when you receive message, you can send back response
49 ## that you've received message
50 self.radio.enableAckPayload()
51
52 ## Because this is a receiving setup we will open Reading pipe
53 # This will be second entry that we created at top
54 self.radio.openReadingPipe(1, self.pipes[1])
55 #It print details
56 self.radio.printDetails()
57 # Start listening for messages
58 self.radio.startListening()
59
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: bash

Roks-MBP:~ rokivancic\$

Ln 48, Col 39 Spaces: 4 UTF-8 LF Python

Slika 4.3: Nastavitve komunikacijskega modula za sprejemanje podatkov



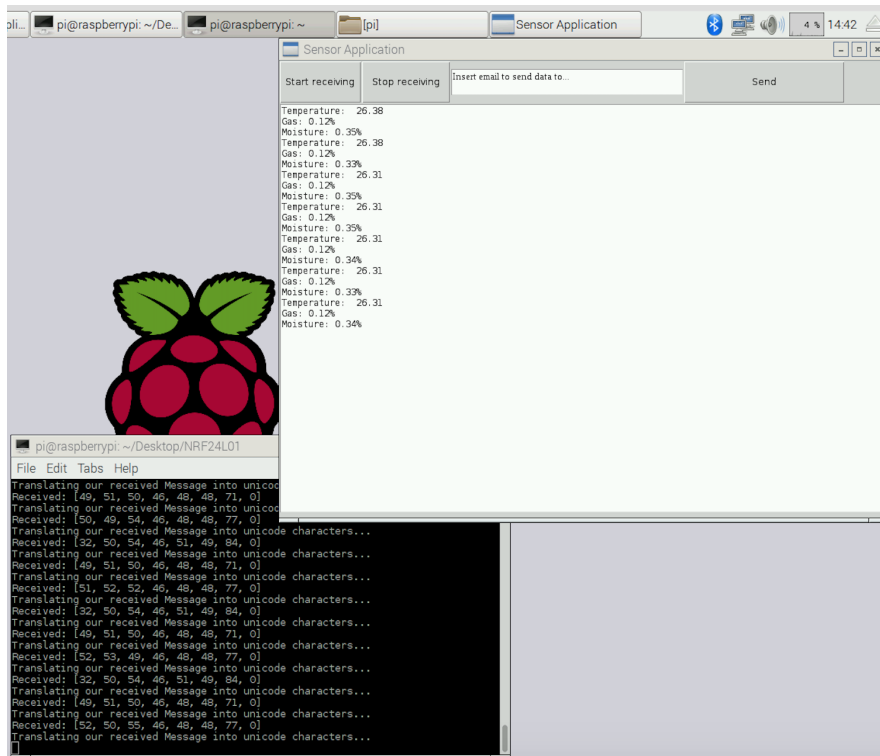
Slika 4.4: Dekodiranje in izpis sporočila

4.1.4 Aplikacijski del

Ker je način branja podatkov v konzolni aplikaciji za navadnega uporabnika prezapleten, smo se odločili izdelavo preproste aplikacije, ki izpisuje meritve ter omogoča njihovo pošiljanje preko elektronske pošte. Aplikacija je napisana v programskem jeziku Python s pomočjo knjižnice *tkinter* in *MIMEText*. Prva knjižnica je namenjena gradnji okenske aplikacije in uporabi vseh gradnikov, kot so gumbi, tekstovna polja in tako naprej [13], druga pa omogoča pošiljanje elektronskih sporočil iz podanega elektronskega naslova.

Program je torej preprosta okenska aplikacija s tremi gumbi (Start receiving, Stop receiving in Send) ter dvema tekstovnima poljema, za vnos

elektronskega naslova ter prikaz prejetih podatkov (Slika 4.5).



Slika 4.5: Aplikacija za izpis ter pošiljanje podatkov

Za pošiljanje elektronskega sporočila se uporablja knjižnica *MIMEText*[4], katera omogoča enostavno uporabo. V zgornji opravilni vrstici aplikacije je polje za vnos elektronskega naslova, kamor lahko vpišemo želeni elektronski naslov in pošljemo senzorske podatke. Ta se zapiše v spremenljivko *receiverEmail*, ki s pomočjo regularnih izrazov preveri format elektronskega naslova. Koda, kjer pridobimo elektronski naslov in zapis kako s pomočjo regularnega izraza preverimo format je podana na sliki 4.6 [10].

```
def sendData(self):
    self.receive = False

    receiverEmail = self.txtReceiver.get("1.0", END)
    if not re.match(r"^[^@]+@[^@]+\.[^@]+", receiverEmail):
        messagebox.showinfo("warning", "Email is not in validate format!")
    else:
        print("Sending data via Mail...")
        server = smtplib.SMTP("smtp.gmail.com:587")
        server.starttls()
        server.login("rok.ivancic92@gmail.com", getPass())
        server.sendmail("rok.ivancic92@gmail.com", receiverEmail, self.messageContent)
        server.quit()
        messagebox.showinfo("Info", "Message was send successfully")
```

Slika 4.6: Dekodiranje in izpis sporočila

4.1.5 Pripomočki

Za potrebe preverjanja toka v vezju smo uporabili multimeter. Naprava omogoča natančno merjenje napetosti in toka v vezju. To nam zagotovi, da imajo senzorji, ki so na to občutljivi, točno določeno vhodno napetost in tok.

Ker gre le za prototip, je bil celoten sistem grajen na testni razvojni ploščici (angl. protoboard). Gre za ploščico, ki nam omogoča enostavno povezovanje različnih komponent v sistemu. Z njo lahko enostavno testiramo in popravljamo, če kje pride do napake. Za nekatere senzorje, ki niso povsem prilagojeni testiranju, pa smo s spajkalnikom povezali žičke z vhodnimi pini tako, da se ti enostavno lahko priklopijo na testno ploščico.

Za lažje pisanje in spremljanje sprememb kodi smo uporabili programsko opremo za verzioniranje Git oziroma BitBucket, ki je zastonjsko okolje za osebno rabo, kjer se lahko spremlja verzije privatnih projektov.

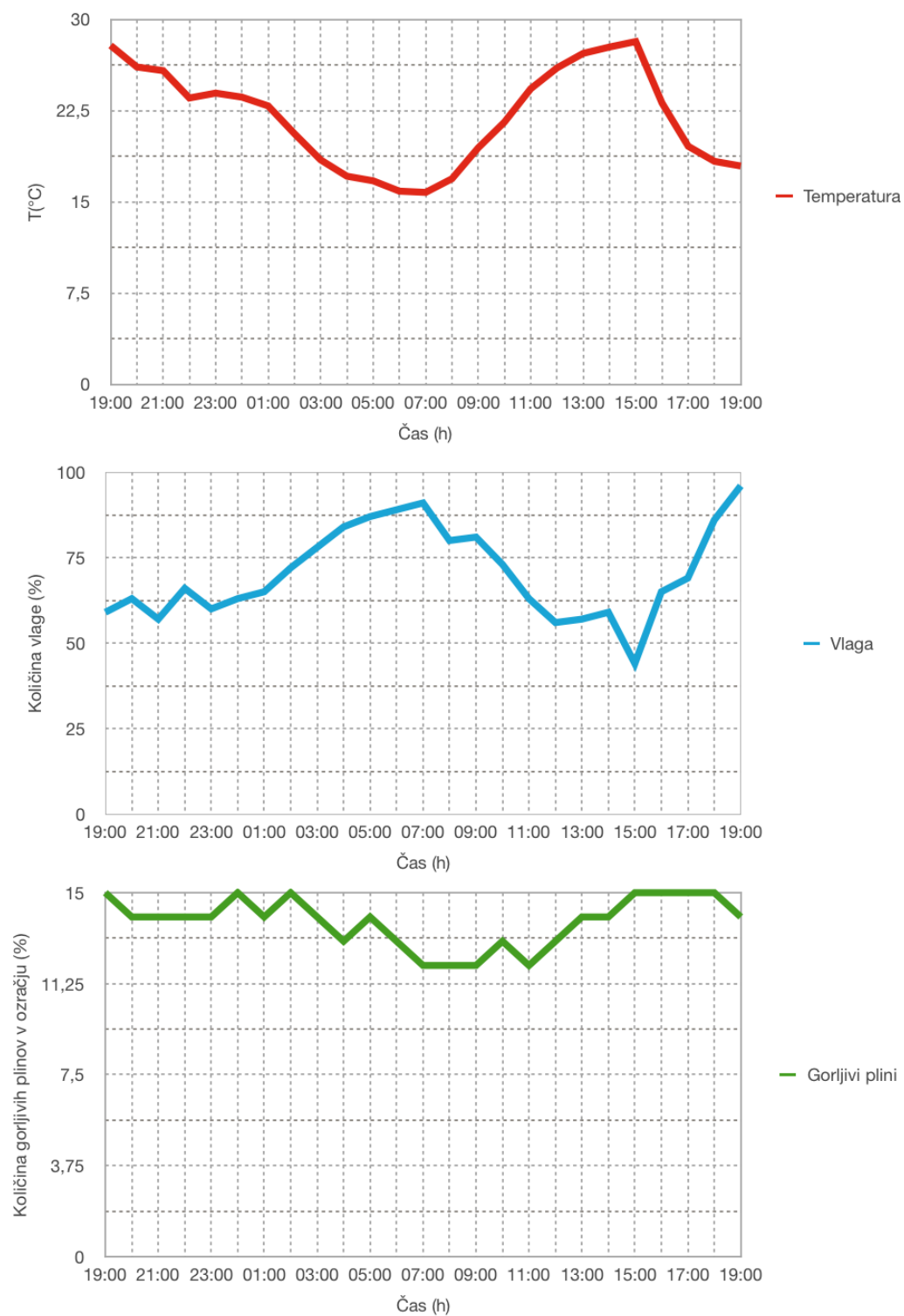
4.2 Primer uporabe

Ko je celoten sistem razvit in pripravljen za uporabo je čas, da preizkusimo in tako še dodatno testiramo naš sistem. Preizkus senzorskega sistema smo izvedli na prostem, kjer smo merili zunanjo temperaturo, vlago ter morebitne gorljive pline. Podoben preizkus smo opravili še v kleti, kjer hranimo nekatera živila. Vsaka od meritev je trajala 24 ur. Sistem je začel meritev izvajati ob 19.00 ter končal naslednji dan ob 19.00. Vsako polno uro se je na Raspberry Pi zabeležila meritev, ki se je na koncu meritve poslala preko elektronske pošte na osebni računalnik v analizo.

4.2.1 Meritev na prostem

Za meritev na prostem smo si izbrali dvorišče pred hišo, kjer sta modula še dovolj blizu za prenos in sprejemanje podatkov. Senzorski sistem je stal v senci ter vsako uro beležil meritve. Zbrani podatki so s pomočjo preglednice predstavljeni kot grafi, za lažjo vizualno predstavitev.

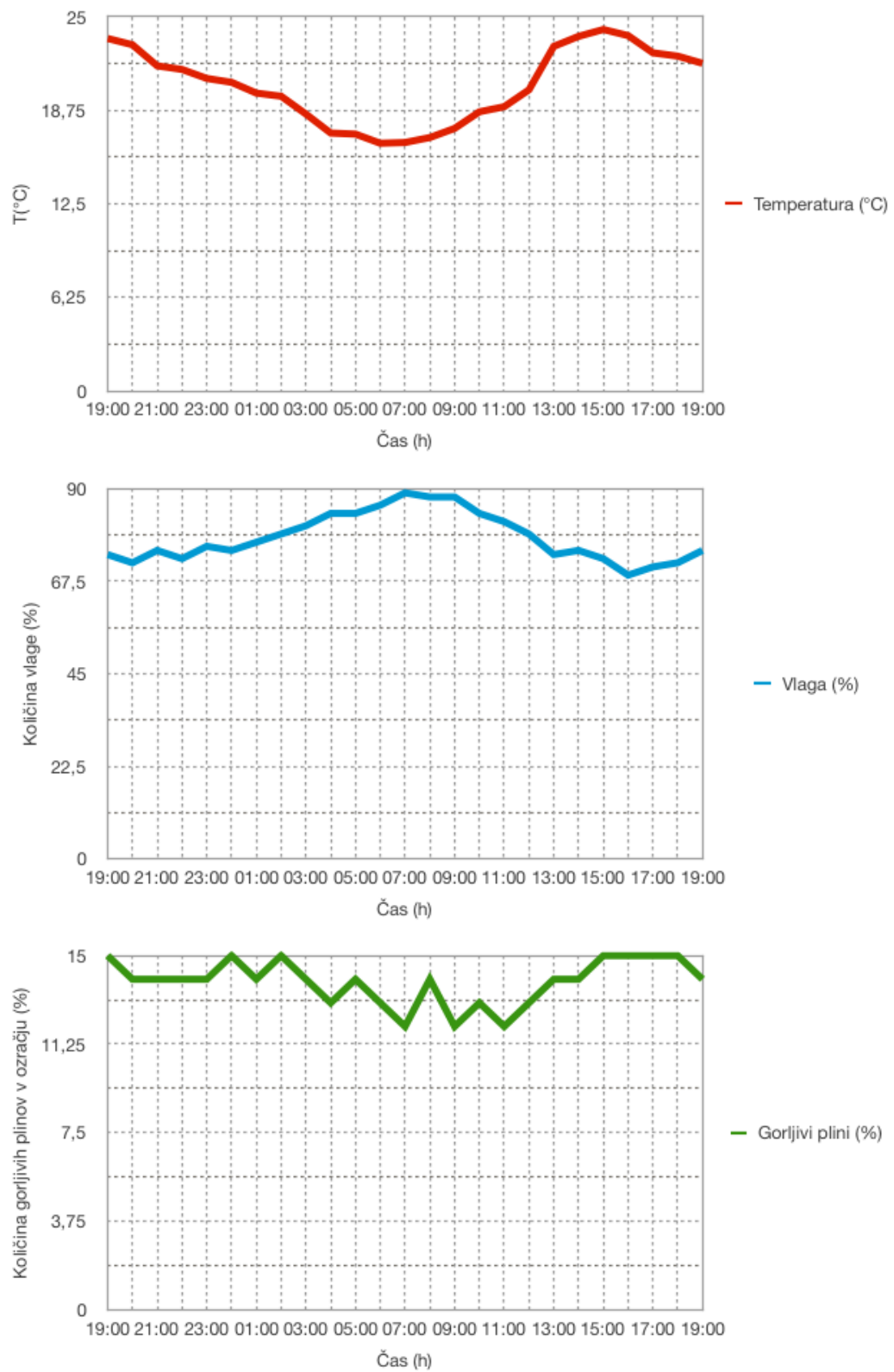
Iz pridobljenih meritev vidimo spremembe v temperaturi in vlagi medtem ko se količina gorljivih plinov ni bistveno spreminjala. Temperatura čez noč pada iz 28°C v popoldanskem času, na slabih 16°C v jutranjih urah, kjer se po 7h prične ponovno dvigovati na povprečno poletno temperaturo. Količina vlage v enakem obdobju ima ravno obraten vzorec pot kot temperatura. Ponoči se količina vlage visoko dvigne, najvišja pa je zjutraj, preden posije sonce. Nekje ob 16h pa vidimo strm padec temperature. Takrat pa se je pričenjalo nevihta, ko je najprej zapihal močan veter, kar se vidi kot padec temperature in vlage. Kmalu zatem se je pričel popoldanski dež v obliki lokalne nevihte. To se dobro vidi na sliki 4.7 oziroma grafu, kjer temperatura strmo pade, vlaga pa strmo naraste. Količina gorljivih plinov je tudi v tem primeru ostala več ali manj enaka, kar je povsem logično.



Slika 4.7: Grafi meritev na prostem, v senci

4.2.2 Meritev v prostoru

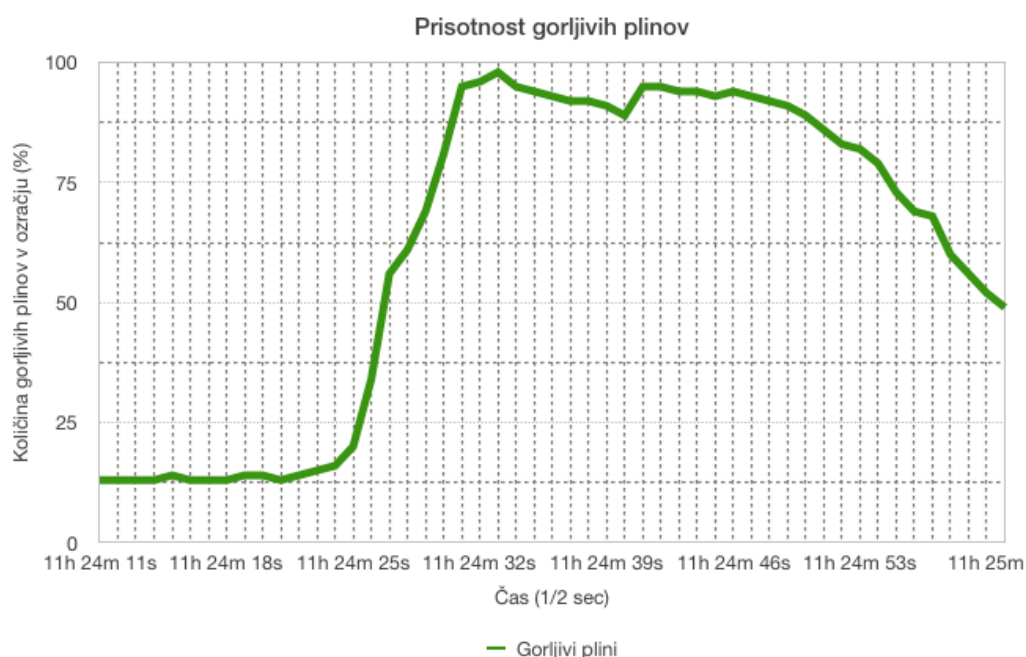
Za meritev v hiši smo si izbrali klet, kjer so prisotna tudi živila, ki jih hranimo. Vrste živil, ki jih hranimo so različna, zanima pa nas le ali je to okolje primerno za shrambo. Pod enakimi pogoji smo ozračje spremljali tudi v kleti in sicer izvajali meritev vsako uro. Klet sicer velja za bolj hladno, a ker je polovico pod zemljo, je v času dežja tudi zelo vlažna, kar je tudi prikazano na sliki 4.8, oziroma grafu. Temperatura čez dan je nižja kot zunaj ter počasneje pada in narašča. Vlaga pa je v tem primeru konstantna in visoko nad povprečjem, ponoči tudi do 90%, kar je za prostor z živili neugodno. Gorljivih plinov tudi v tem primeru ni bilo. Iz tega lahko izhajamo, da v tem času v kleti ni pametno hraniti živil, katere zahtevajo suho in mrzlo ozračje, lahko pa so živila, katera so dobro zapakirana ter jih nekoliko višje temperature ne motijo.



Slika 4.8: Grafi meritev v prostoru, klet

4.2.3 Meritev plinov

Ker prisotnost plinov v obeh primerih ni znašala več kot 15%, smo test gorljivih plinov izvedli drugače. Senzorski sistem je nastavljen tako, da zajema podatke vsako sekundo ter ob koncu enako kot prej, pridobljene podatke pošlje uporabniku. V bližini senzorja smo s pomočjo vžigalnika preizkusili senzor tako, da smo zanetili majhen košček papirja oziroma tudi sam vžigalnik je bil v tem primeru dovolj, ter postavili v bližino senzorja. Na sliki 4.9 je prikazana koncentracija plinov v trenutku, ko zanetimo ogenj v bližini senzorja. Odzivnost zaznavanja plinov je v tem primeru primerna, saj je zaznavala že ko se je iz vžigalnika gorljivi plin vžgal. Ker se je poskus izvajal v zaprtem prostoru, je padec gorljivih plinov manjši, kot če bi bili na prostem.



Slika 4.9: Graf meritve plinov v prostoru

Na sliki 4.10 je prikazana meritev plina, katera je bila preko elektronske pošte poslana na naš naslov. Poleg so tudi meritve za vlago in temperaturo.

Start Time: 30/08/2017
Log Interval (s): 0.5
Gas (%) , Humidity (%) , Temperature (C)

13, 0.61, 26.43
13, 0.61, 26.43
13, 0.61, 26.43
13, 0.61, 26.43
14, 0.61, 26.43
13, 0.61, 26.43
13, 0.61, 26.43
13, 0.61, 26.43
14, 0.61, 26.43
14, 0.61, 26.40
13, 0.61, 26.40
14, 0.61, 26.40
15, 0.60, 26.40
16, 0.60, 26.40
20, 0.61, 26.40
34, 0.61, 26.40
56, 0.61, 26.40
61, 0.61, 26.40
69, 0.61, 26.40
81, 0.61, 26.40
95, 0.61, 26.40
96, 0.61, 26.40
98, 0.61, 26.40
95, 0.61, 26.43
94, 0.61, 26.43
93, 0.61, 26.43
92, 0.62, 26.43
92, 0.62, 26.43
91, 0.62, 26.43
89, 0.62, 26.43
95, 0.62, 26.43
95, 0.62, 26.43
94, 0.62, 26.43
94, 0.62, 26.43
93, 0.62, 26.43
94, 0.62, 26.43
93, 0.62, 26.43
92, 0.62, 26.43
91, 0.62, 26.43
89, 0.62, 26.43
86, 0.62, 26.43
83, 0.62, 26.43
82, 0.62, 26.43
79, 0.62, 26.43
73, 0.61, 26.43
69, 0.61, 26.43
68, 0.61, 26.43
60, 0.61, 26.43
56, 0.61, 26.43
52, 0.61, 26.41
49, 0.61, 26.41

Slika 4.10: Prejeti podatki o meritvi

Poglavje 5

Sklepne ugotovitve

V diplomski nalogi smo uspešno izdelali prototip senzorskega sistema za merjenje temperature, vlage in gorljivih plinov. Opredelimo ga lahko kot primer rešitve na področju IoT, ki je ta trenutek v porastu in daje človeku možnost, da si zgradi napravo, ki mu bo koristila v določenem okolju. Naš sistem sestoji iz več komponent, kot so Arduino, Raspberry Pi, senzorji za temperaturo, vlago ter gorljive pline in dva komunikacijska modula za brezžično komunikacijo med mikrokrmilnikoma. Končni rezultat je tabela meritev, ki smo jih pridobili zunaj na prostem, v senci ter v hiši, natančneje v kletnem prostoru. Iz rezultatov smo tudi lahko sklepali, da je kletni prostor primeren le za živila, ki so dobro pakirana, kjer jim vlaga ne more pokvariti roka ali vplivati na njihovo kakovost.

Pri gradnji sistema smo naleteli na nekaj težav, ki so upočasnile razvoj. Prva težava se je pojavila pri združevanju vseh senzorskih komponent in komunikacijskega modula. Arduino za izhodno napetost ponuja pin z napetostjo 3.3 in 5V. Izhodna napetost 3.3V je uporabljena za komunikacijski modul, medtem ko je 5V napetost uporabljena za napajanje razvijalne ploščice na kateri so ostali tri senzorji. Seveda so vsi senzorji zaradi premajhne napetosti delovali nepravilno. Problem smo rešili tako, da smo kupili dodatne baterije, ter jih zaporedno zvezali tako, da je bila izhodna napetost baterij primerna za dani senzor. Podoben problem smo imeli pri postavitvi sistema

za merjenje ozračja, kjer ni bilo električnega priklopa za napajanje senzorskega sistema. V ta namen smo najprej uporabili predpisano 9V baterijo, ki pa je bila še vedno premalo za senzorja, ki sta bila priklopljena na Arduino. V specifikacijah smo zasledili, da je mogoče uporabiti tudi 12V baterijo, kjer smo z novo rešitvijo dosegli želeno moč za optimalno delovanje senzorjev.

Izdelan sistem je daleč od popolnega, saj bi se dalo sistem marsikje nadgraditi.

- Ena izmed nadgradenj bi bilo filtriranje gorljivih plinov iz senzorja, ki nam omogoča izhodne podatke prebrati tako, da podajajo vsak plin posebej. Tako bi lahko natančno videli količino vseh plinov, katere senzor podpira.
- Druga nadgradnja je v aplikaciji na Raspberry Pi. Trenutna rešitev je funkcionalno minimalistična, s katero beremo podatke, ter jih preko elektronske pošte pošljemo na zeleni elektronski naslov. Aplikacijo bi lahko nadgradili tako, da bi podatke predstavili v preglednici ali da bi omogočili gumb za shranjevanje podatkov, urejanje in tako naprej.
- Pomembna nadgradnja je tudi brezžična komunikacija. Komunikacijska modula v diplomski nalogi zmoreta komunicirati do 100m na prostem ter približno 30-50m v zaprtih prostorih. Nadgradili bi jih lahko z anteno, katero se lahko dokupi kot dodatno opremo modula. Tako bi lahko komunicirali tudi do 200m na prostem.

Literatura

- [1] Arduino Software (IDE) [online]. Dosegljivo: <https://www.arduino.cc/en/Guide/Environment>.
- [2] Arduino Wireless Communication – NRF24L01 [online]. Dosegljivo: <http://howtomechatronics.com/tutorials/arduino/arduino-wireless-communication-nrf24l01-tutorial/>.
- [3] DIY Soil Testing with Arduino and FC-28 Moisture Sensor [online]. Dosegljivo: <https://diyhacking.com/arduino-soil-moisture-sensor/>.
- [4] Email: Examples [online]. Dosegljivo: <https://docs.python.org/2/library/email-examples.html>.
- [5] Feeding power to Arduino: the ultimate guide [online]. Dosegljivo: <https://www.open-electronics.org/the-power-of-arduino-this-unknown/>.
- [6] Hladna veriga za zagotavljanje varnosti živil [online]. Dosegljivo: <http://www.nijz.si/sl/hladna-veriga-za-zagotavljanje-varnosti-zivil>.
- [7] Internet of things - tutorialspoint [online]. Dosegljivo: https://www.tutorialspoint.com/internet_of_things/internet_of_things_tutorial.pdf.

-
- [8] Logistics 2017: connecting the dots between IT systems to make sense of the data [online]. Dosegljivo: <https://dyzle.wordpress.com>.
 - [9] Raspberry Pi [online]. Dosegljivo: https://en.wikipedia.org/wiki/Raspberry_Pi.
 - [10] Regular expression operations [online]. Dosegljivo: <https://docs.python.org/2/library/re.html>.
 - [11] SGas Sensor(MQ2) [online]. Dosegljivo: http://wiki.seeed.cc/Grove-Gas_Sensor-MQ2/.
 - [12] Smoke Detection using MQ-2 Gas Sensor [online]. Dosegljivo: <https://create.arduino.cc/projecthub/Aritro/smoke-detection-using-mq-2-gas-sensor-79c54a>.
 - [13] Tkinter — Python interface to Tcl/Tk [online]. Dosegljivo: <https://docs.python.org/2/library/tkinter.html>.
 - [14] TMP36 temperature sensor [online]. Dosegljivo: <https://learn.adafruit.com/tmp36-temperature-sensor/using-a-temp-sensor>.
 - [15] What is an Arduino? [online]. Dosegljivo: <https://learn.sparkfun.com/tutorials/what-is-an-arduino>.
 - [16] What is Arduino? [online]. Dosegljivo: <https://www.arduino.cc/en/Guide/Introduction>.
 - [17] What is the internet of things? [online]. Dosegljivo: <https://www.theguardian.com/technology/2015/may/06/what-is-the-internet-of-things-google>.
 - [18] Wireless Remote Using 2.4 Ghz NRF24L01 : Simple Tutorial Using of NRF24L01 and Arduino [online]. Dosegljivo: <http://www.instructables.com/id/Wireless-Remote-Using-24-Ghz-NRF24L01-Simple-Tutor/>.